

AD A139959

Semiannual Technical Summary

Restructurable VLSI Program

30 September 1983

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

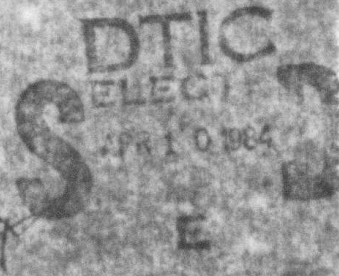
LEXINGTON, MASSACHUSETTS



Prepared for the Defense Advanced Research Projects Agency
under Electronic Systems Division Contract F19628-80-C-0002.

Approved for public release; distribution unlimited.

DTIC FILE COPY



84 04 09 030

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored by the Defense Advanced Research Projects Agency under Air Force Contract F19628-80-C-0002 (ARPA Order 2797).

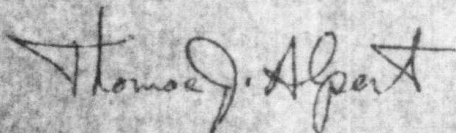
This report may be reproduced to satisfy needs of U.S. Government agencies.

The views and conclusions contained in this document are those of the contractor and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the United States Government.

The Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER



Thomas J. Alpert, Major, USAF
Chief, ESD Lincoln Laboratory Project Office

Non-Lincoln Recipients

PLEASE DO NOT RETURN

Permission is given to destroy this document
when it is no longer needed.

**MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY**

RESTRUCTURABLE VLSI PROGRAM

**SEMIANNUAL TECHNICAL SUMMARY REPORT
TO THE
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY**

1 APRIL — 30 SEPTEMBER 1983

ISSUED 6 FEBRUARY 1984

Approved for public release; distribution unlimited.

LEXINGTON

MASSACHUSETTS

ABSTRACT

This report describes work performed on the Restructurable VLSI Program sponsored by the Information Processing Techniques Office of the Defense Advanced Research Projects Agency during the period 1 April 1983 through 30 September 1983.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



TABLE OF CONTENTS

Abstract	iii
List of Illustrations	vi
List of Tables	vi
 I. PROGRAM OVERVIEW AND SUMMARY	 1
A. Overview	1
B. Summary of Progress	2
1. Design Aids for RVLSI	2
2. Applications	3
3. Testing	3
4. RVLSI Technology	4
 II. DESIGN AIDS FOR RVLSI	 5
A. MACPITTS	5
B. Lincoln Boolean Synthesizer (LBS)	6
C. Chip Assembler	7
1. Cell Definition and Placement	7
2. Global Routing	7
3. Channel Routing	8
D. Restructurable Wafer Editor (RWED)	10
 III. APPLICATIONS	 11
A. Dynamic Time Warping System	11
B. Digital Integrator	19
 IV. TESTING	 21
A. Tester-on-Chip (TOC) Development	21
B. Experience with the Optical Probe	22
 V. RVLSI TECHNOLOGY	 25
 References	 29

LIST OF ILLUSTRATIONS

Figure No.		Page
1.	Channel with Vertical Constraints and Constraint Loop	9
2.	DTW Wafer with 63 Processing Elements	12
3.	DTW Processing Element (PE)	14
4.	Distance Computer	16
5.	One Stage (MA) of Distance Computer	17
6.	Path Computer	18
7.	Digital Integrator Yields	20
8.	Photomicrographs of Clock Distribution Wires and Pad Wiring Channels in Defective TOC Circuits	21
9.	Current Paths for Laser Probing of a CMOS Inverter: (a) N-Channel Conducting, (b) P-Channel Conducting	23
10.	Symbolic Representation of 4-Bit ALU Slice Layout	26

LIST OF TABLES

Table No.		Page
I	Comparison of Ordering Methods	6

RESTRUCTURABLE VLSI PROGRAM

I. PROGRAM OVERVIEW AND SUMMARY

A. OVERVIEW

The main objective of the Lincoln Restructurable VLSI Program (RVLSI) is to develop design methodologies, architectures, design aids and testing strategies for implementing wafer-scale systems with complexities approaching a million gates. In our approach, we envisage a modular style of architecture comprising an array of cells embedded in a regular interconnection matrix. Ideally, the cells should consist of only a few basic types. The interconnection matrix is a fixed pattern of metal lines augmented by a complement of programmable switches or links. Conceptually, the links could be either volatile or nonvolatile. They could be of an electronic nature, such as a transistor switch, or could be permanently programmed through some mechanism such as a laser. The RVLSI Program is currently focusing on laser-formed interconnect.

The link concept offers the potential for a highly flexible, restructurable type of interconnect technology that could be exploited in a variety of ways. For example, logical cells or subsystems found to be faulty at wafer-probe time could be permanently excised from the rest of the wafer. The flexible interconnect could also be used to circumvent faulty logic and tie in redundant cells judiciously scattered around the wafer for this purpose. Also, the interconnect could be tailored to a specific application in order to minimize electrical degradations and performance penalties caused by unused wiring and links.

Further, the testing of a particular logical subsystem buried deep within a complex wafer-scale system poses a very difficult problem. A properly designed restructurable interconnect matrix could be temporarily configured to improve both the controllability and observability of internal cells from the wafer periphery. In this way, each component cell or a manageable cluster of cells could be tested in straightforward manner using standard techniques. With an electronic linking mechanism, it is possible to think in terms of a dynamically reconfigurable system. Such a feature could be used to alter the functional mode of a system subject to changes in the operating scenario, or it could be used to support some degree of fault tolerance if the system architecture was suitably designed.

Several major areas of research have been identified in the context of the RVLSI concept:

- (1) System architectures and partitioning for whole-wafer implementations.
- (2) Placement and routing strategies for optimal utilization of redundant resources and efficient interconnect.

- (3) Assignment and linking algorithms to exploit redundancy and flexible interconnect.
- (4) Methods for expediting cell design with emphasis on functional level descriptions, enhanced testability, and fault tolerance.
- (5) Methods for testing complex, multiple-cell, whole-wafer systems.

Complementary work on the development of various link and interconnect technologies as well as fabrication/processing technology is being supported by the Lincoln Air Force Line Program, and results are reported under the Lincoln Laboratory Advanced Electronic Technology Quarterly Technical Summary.

B. SUMMARY OF PROGRESS

Work for this period is reported under four headings: Design Aids for RVLSI (Section II), Application (Section III), Testing (Section IV), and RVLSI Technology (Section V). The following paragraphs summarize progress to date.

1. Design Aids for RVLSI

During the last reporting period, a new Weinberger array column-ordering algorithm was implemented and integrated into the MACPITTS silicon compiler. This is a recursive min-cut technique based on a method suggested by Fiduccia and Matheyses. This was done to improve the run time and performance of the earlier ordering system which was based on a classical, exponential-time, hill-climbing approach. A substantial amount of data was taken, using a number of sample MACPITTS circuit designs, comparing the min-cut to the hill climber and to a combination system where the min-cut was used as a preprocessor for the hill climber. It was concluded that the combination system represented the best choice, particularly for large designs with a nominal 2:1 improvement in run time and slightly better packing efficiency than either approach alone could offer.

The Lincoln Boolean Synthesizer has been upgraded to incorporate the Weinberger array min-cut ordering routine described above. Automatic power bus sizing is also now supported.

The first version of a technology-independent, hierarchical chip-assembly tool, which supports manual placement and automatic routing for implementing complex circuit designs from an arbitrary number of simpler substructures, is now operational. The routing problem is addressed in three major steps where the component cells are first manually placed using a CAESAR-based graphics interface; a 'loose' or 'global' routing is next performed based on free area available; and, finally, a channel router is invoked to form the final interconnect within each free area. The system is undergoing test and evaluation to study the effectiveness of the algorithms being used.

The restructurable wafer editor (RWED) which forms the software controller interface between the VAX host facility and the laser programming station has been upgraded to incorporate more sophisticated compensations for mapping virtual coordinates to physical wafer positions. This has been made necessary due to laser table positioning inaccuracies which manifest themselves over the full extent of a 3-inch wafer.

2. Applications

The detailed logical architectural definition of the dynamic time-warping wafer has progressed significantly during the last reporting period. The Myers/Rabiner level building algorithm has been selected as the basic methodology for connected-word recognition. Changes in the basic architectural approach have made it possible to incorporate some additional, run-time-programmable flexibility. Current plans call for supporting the Itakura-Saito log-likelihood metric and sum-of-magnitude-differences (Chebyshev) distance measure. Cell design is well underway on the more mature processing elements. A detailed functional simulation has been developed in the C programming language and is being used to verify the logical correctness of the design. Additionally, a higher level simulation embracing the entire recognition system has been put in place using high-performance, in-house-developed, programmable processors to reduce run time.

The phase I digital integrator effort is entering the final phase. During the last reporting period, all yields at first level metal have routinely exceeded 50%, well beyond the 33% needed. Unexpected attrition between first- and second-level metal was traced to inadequate coverage of polysilicon contacts during amorphous silicon etch. Mask changes have been implemented to correct the problem. Partially populated test wafers are currently being programmed. Also, a 68000-based, test-while-zap system has been completed and interfaced to the VAX.

3. Testing

Thirteen tester-on-chip (TOC) circuits were returned from the M33M 3 μ M NMOS run at the end of June. They were tested on the Tektronix 3260 facility using a test vector file generated by the switch level simulator. None were found functional, and optical inspection revealed poor metal patterning. The design was submitted for the M37A run, chips from which are currently undergoing evaluation.

A prototype optical-probe capability for CMOS has been implemented consisting of a low-power optical laser, a microscope with stage, an oscilloscope, and a simple current-monitoring circuit. A TV monitor is included as an option. This system has been used to examine faulty test circuits generated by LBS. In a 74181-equivalent circuit, a 30:1 ratio of photocurrent was observed in the power rails between the one and zero states of properly functioning gates. On defective gates, the ratio was markedly reduced. Although defective nodes have been successfully identified, more work is needed to quantify the exact failure mechanisms.

4. RVLSI Technology

A 64×4 gate-restructurable Weinberger array has been chosen as the demonstration vehicle for a polyimide-based, lateral-geometry, laser-fusible link. Such a link can be realized with only a single-level metal-processing capability, and therefore should be compatible with the MOSIS foundry, the polyimide being added by Lincoln. When pyrolyzed, the links exhibit 1- to $2\text{-k}\Omega$ resistances. A 74181 4-bit universal ALU has been laid out on the array as a test case. A first attempt at actually programming the array with a laser has not proven fully successful, with some links consistently showing abnormally high resistances.

Steps are being taken to alleviate asymmetries due to layout and mask alignment peculiarities. Also, a variable aperture has been added to the laser making it possible to generate ellipsoidal laser beam burn spots. This shape is much better matched to the lateral geometry of the link.

II. DESIGN AIDS FOR RVLSI

A. MACPITTS

A significant improvement made to the MACPITTS silicon compiler [1] during the second half of FY 83 was the inclusion of a new Weinberger array column-ordering mechanism. This was done to improve compilation speed for evolving designs. The new ordering routine is a recursive min-cut algorithm based on work by Fiduccia and Matheyses [2].

The old hill-climbing algorithm has two problems. First, it is an exponential time algorithm, dependent on the number of columns and the number of nets connecting the columns. Second, like any hill-climbing method, it can get caught at a local minimum and will fail to locate the global minimum.

The min-cut algorithm tries to divide the set of columns into two parts of approximately equal 'size', minimizing the number of nets ('cuts') crossing the boundary. The size of a side is calculated by totaling the sizes of every column on that side. The size of a column is the same as the number of nets to which it belongs. By equalizing the sizes, there is a guarantee that the number of tracks needed on each side should be approximately the same. Another interesting point is that to avoid local minima, the algorithm explores cases where the situation temporarily gets worse.

This method is tried recursively on the array of columns. It starts with all of the columns, dividing them into two groups. Then the min-cut is used on each half, dividing both into two groups. This continues recursively until the number of columns in a group is reduced to four. (This number is called the 'leaf number' of columns and is adjustable. It might be interesting to investigate the effect of varying this parameter.) It can be seen that the number of calls to the min-cut is linear with the number of columns. Actually, it is directly proportional to the quotient of the number of columns divided by the leaf number of columns.

After the min-cut was installed and tested, it was decided to try to order columns in two passes. The min-cut would perform a fast rough ordering. It would be followed by the more exhaustive hill-climbing ordering routine. It was thought that this combination would yield better results and tests have confirmed this hypothesis. Table I compares the three ordering methods for various MACPITTS designs. The resulting number of tracks (the variable being minimized) and the running times are listed.

From this table, we have concluded that the combination of the min-cut and hill-climbing methods gives the best result, in time comparable to or better than the hill-climbing method alone. With large designs (e.g., frisc 16 with 272 columns), the combination runs in a little over half the time of the old version with better results. The min-cut alone does well on small designs. On larger designs, it runs faster and gets close to the number of tracks that the hill-climber calculates.

TABLE I
Comparison of Ordering Methods

Project	Tracks					Time in Minutes		
	Columns	None	Min	Hill	Both	Min-cut	Hill	Both
addr	17	10	5	5	4	0.36	0.53	0.60
agc	119	66	47	42	40	24.73	40.95	41.39
bit2	36	21	14	12	13	1.62	2.47	2.42
compare	6	2	2	2	2	0.06	0.07	0.12
corr	16	4	4	4	4	0.16	0.21	0.31
counter	11	7	6	4	4	0.19	0.30	0.37
frisc16	272	159	102	92	86	392.50	966.60	533.00
parity	22	18	8	8	6	0.68	1.04	1.16
rshift	33	21	14	12	12	1.29	3.46	2.37
rshift2	18	10	8	6	3	0.34	0.51	1.08
shifter	14	10	8	8	8	0.27	0.22	0.41
srff	6	4	4	3	3	0.03	0.07	0.13
taxi	55	31	26	24	18	2.81	4.29	7.47
toc	283	155	106	-	96	158.50	-	286.00

A better way to compare the ordering methods would be to construct a complexity measure, then measure the complexity of many problems. These problems then could be ordered, comparing the ordering results for each method. The complexity measure could comprise such parameters as the number of nets, the average number of nets per column and the number of fixed columns.

The next planned improvement to MACPITTS is the incorporation of the channel router (see Section II-C).

B. LINCOLN BOOLEAN SYNTHESIZER (LBS)

Two new features have been added to the LBS system previously reported [3]:

- (1) A min-cut ordering program has been incorporated into LBS (see Section II-A). This heuristic orders the columns of the array trying to reduce the number of nets crossing imaginary 'cut lines'. These are lower bounds for the number of horizontal tracks necessary to pack the signals in the array.
- (2) Automatic power bus sizing is now supported, calculating the width of the voltage and ground lines as a function of the number of columns in the array.

C. CHIP ASSEMBLER

The Chip Assembler is a manual placement/automatic routing system for the implementation of complex integrated circuits from an arbitrary number of simpler substructures, based on a hierarchy of cells. A first version of the system is now in operation and is undergoing testing with a variety of examples to study the effectiveness of the algorithms employed, discover any hidden problems, and make the appropriate corrections and improvements.

The approach taken to the general routing problem is to divide it into several phases that can be treated independently. Each of these phases presents us with a simpler problem that we can solve successively, paying an acceptable price for the losses inherent in the partitioning process.

The routing problem is treated in three steps: the cells are manually placed, a 'loose' or 'global routing' [4] is performed based on the free areas available, and finally a channel router is invoked to perform the final interconnect in each free area.

Accordingly, the system consists of three main parts:

1. Cell Definition and Placement

CAESAR, a graphics editor developed at the University of California-Berkeley [5] is used to enter the necessary information about the basic cells, their placement and the 'bounding box' for the new cell. This data is then automatically converted into our formats.

The information about the cells is kept to a minimum. Only their outlines, positions of their inputs and outputs, and locations are used by the Chip Assembler to produce the new cell. We note that the cells can have arbitrary rectilinear shapes, and are not restricted solely to rectangles.

The net list describing the required connections is given in a separate text file. These nets can be arbitrary multi-pin nets.

2. Global Routing

The free area in the cell is divided into a set of rectangles, called channels, that may intersect only along their boundaries. The global routing process consists of placing 'imaginary pins' (as opposed to the 'real pins' of the cells), called crossing pins, on the intersection of the channels so that by completing the interconnections inside each channel, all the nets are routed.

This process goes on in several steps:

- (a) For each net, we select a collection of paths based on a minimum distance criteria. A path is simply an ordered collection of channels that we can cross (i.e., they intersect in pairs) so that all the pins in the net belong to a channel. This selection is done converting the channel information into a graph,

assigning a distance (based on the Manhattan distance) to the edges and using a minimum distance algorithm to find spanning trees.

The basic graph is modified for each net to take into account the position of its pins in the channels.

The paths obtained for each net are ordered according to their cost.

- (b) We now select one path for each net so that there is enough space at each channel crossing to place the necessary number of pins for all the nets.

When selecting between the paths of one net, their costs are modified considering the amount of available space remaining at each crossing, contributing in this way to spreading the crossing points.

We pick a path for the net based on these updated costs. If no path can be selected because there is not space available at a crossing, limited backtracking is attempted. We look for a net already allocated, whose path is blocking the present net, and we try to select a different path to make free space.

- (c) Finally, we determine the exact location for the crossing pins. As we already know that there is enough space at each crossing to place all the pins, this can be looked upon as selecting permutations for each of the pins at each crossing. Two points are taken into consideration to do this allocation. The simpler is a 'local' rule: place the pin against the same side of the opening as the pin of the net already in the channel (or some average for multiple pins), and start moving it in the opposite direction until it doesn't violate clearance rules. This tends to minimize the length of the interconnections. The other is a 'global' rule: look ahead for possible crossings along several channels and try to place as many pins as possible simultaneously along a straight line, minimizing the number of joggings that the channel router will have to produce. Both criteria are implemented at this time.

3. Channel Routing

Finally, each channel and its net list are given to a channel router based on an algorithm presented in [6]. The Chan algorithm is a two-layer router that can handle multiple pin nets. The pins can be located in specific places on two opposite sides of the channel (usually termed the 'top' and 'bottom' of the channel) and channel exit pins located someplace along the other two sides, depending on the track to which the net is eventually assigned. The number of tracks, and hence the channel size, can grow to the amount required to route the channel.

The router in the Chip Assembler was modified to handle the restriction that the channel is required to maintain the same size as originally defined by the CAESAR placement. Also, pins are specifically placed along any of the four sides of the channel and on any of the two possible routing layers.

135010-N

-

- Figure 1. Channel with vertical constraints and constraint loop.**

- 9

D. RESTRUCTURABLE WAFER EDITOR (RWED)

The program which maps design space into physical locations on the laser table is being revised. The present program provides for a two-dimensional linear fit to compensate for wafer distortions, laser table runout, etc. In order to improve registration on 3-inch wafers, particularly at the wafer periphery, and reduce the probability of laser zapping of polyimide around metal cut areas, a transformation using second and higher order terms has been written and registration experiments are in progress.

Initially, the transformation from virtual to physical space was:

$$x_{\text{phys}} = a_0 + a_1 x_{\text{virt}} + a_2 y_{\text{virt}}$$

$$y_{\text{phys}} = b_0 + b_1 y_{\text{virt}} + b_2 x_{\text{virt}}$$

This provided for an arbitrary translation, rotation, and linear stretch of the axes. The new transformation is:

$$x_{\text{phys}} = a_0 + a_1 x_{\text{virt}} + a_2 y_{\text{virt}} + a_3 x_{\text{virt}}^2 + a_4 x_{\text{virt}} y_{\text{virt}} \\ + a_5 y_{\text{virt}}^2 + a_6 x_{\text{virt}}^3$$

$$y_{\text{phys}} = b_0 + b_1 y_{\text{virt}} + b_2 x_{\text{virt}} + b_3 y_{\text{virt}}^2 + b_4 y_{\text{virt}} x_{\text{virt}} \\ + b_5 x_{\text{virt}}^2 + b_6 y_{\text{virt}}^3$$

For each new alignment point on the wafer used by the operator, a corresponding term is added to the series. Each alignment point represents a correspondence between virtual and physical space, and generates an equation of the above form. The resulting system of equations is solved for the 'a's and 'b's, which in turn are used to map from virtual to physical space.

III. APPLICATIONS

A. DYNAMIC TIME WARPING SYSTEM

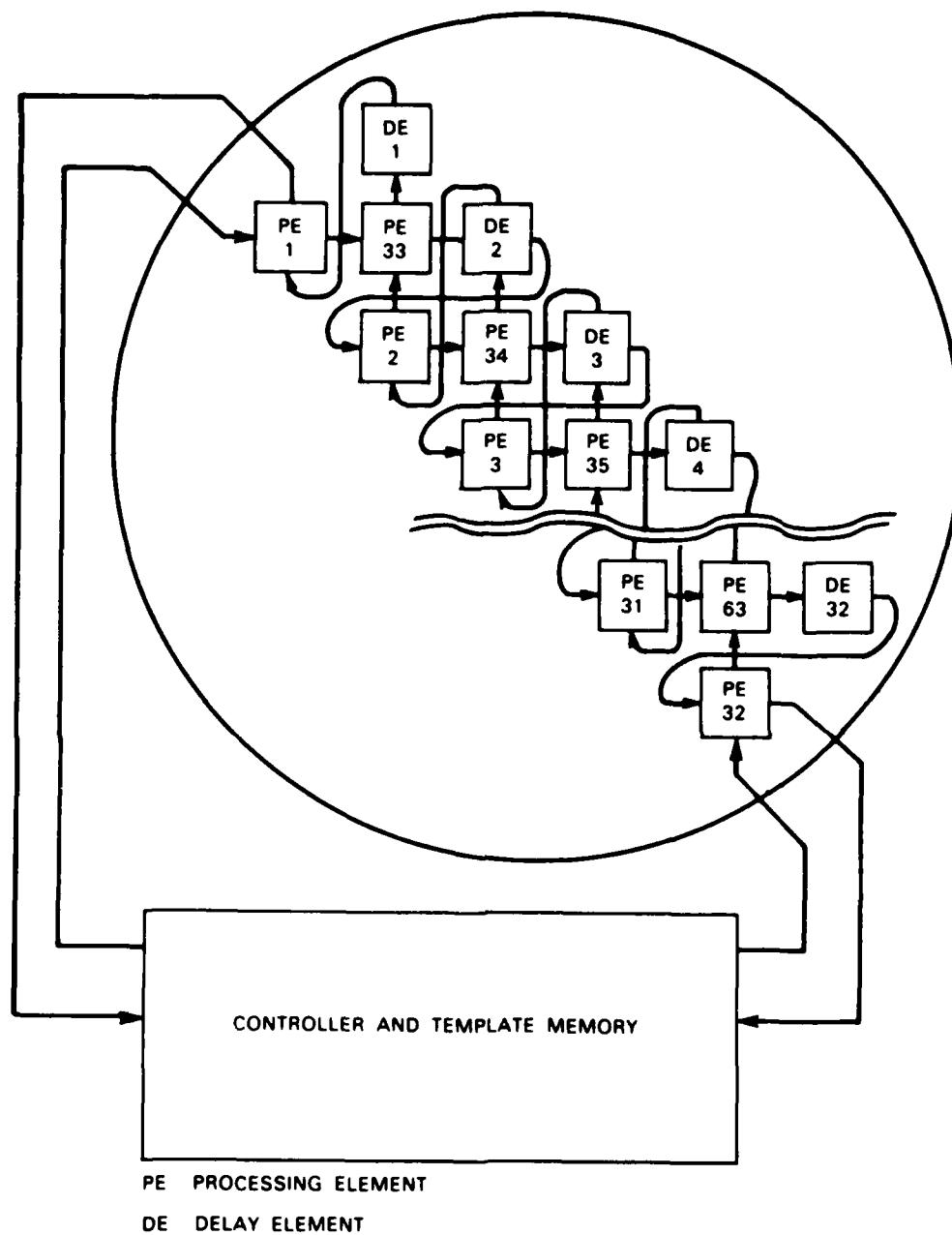
The dynamic time warping (DTW) wafer is intended to provide a powerful engine for the computationally intensive tasks of template matching and dynamic time warping in isolated and connected word-recognition systems. The DTW wafer will be fabricated using the RVLSI wafer-scale technology on a 3-inch wafer with 5 μ M bulk CMOS design rules. The system architecture consists of 63 bit-serial processing elements and 32-bit delay elements connected in a systolic array. The processing and delay elements are composed of four restructurable cell types which will be imaged on the wafer with a 2:1 circuit redundancy to achieve a wafer-scale system comprising over 300,000 functioning transistors.

During the last reporting period, progress on the DTW wafer has been made in several areas. The Myers/Rabiner (Bell Laboratories) level-building algorithm [7] has been chosen as the basic methodology for connected word recognition. Changes in the DTW wafer architecture have been made resulting in an increase in the run-time programmability and flexibility of the system. The detailed logical/architectural definition of the DTW system is well advanced. The more mature portions of the architecture have been transitioned to integrated circuit layout. A detailed logical simulation of the wafer system has been written in the C programming language and is being used for verification of the logical correctness of the design. A higher level simulation of the entire connected word algorithm has been put in place employing the high speed Lincoln Digital Signal Processor (LDSP) real time facility.

The DTW wafer architecture is being tailored to implement the Myers/Rabiner (Bell Laboratories) Level Building Dynamic Time Warping algorithm for connected word recognition [7]. This is achieved by using the DTW wafer as a 'hardware subroutine' called by a more general-purpose higher-level processor which has a lower computational burden. (By default, the wafer can also be used for isolated word recognition.) The key features of the level building algorithm are:

- (1) the algorithm does not need to detect silence between words in a phrase for parsing purposes,
- (2) isolated words can be used for reference template training,
- (3) syntactic constraints can be easily enforced by the higher-level processor without impinging on the special-purpose DTW wafer design,
- (4) for an added cost in computation, speaker-independent systems can be achieved with error rates comparable to speaker-trained systems,
- (5) the word length of an incoming string does not have to be known to the system beforehand.

The algorithm has been demonstrated by Bell Laboratories to achieve digit string recognition accuracies of about 95-96 percent for variable length strings for both speaker-trained and speaker-independent systems.



PE PROCESSING ELEMENT
DE DELAY ELEMENT

Figure 2. DTW wafer with 63 processing elements.

128644-N

In the speech recognition community, a variety of distance metrics and parametric representations have been used. Furthermore, there has been no definitive study on which of these metrics or representations are superior. Since the DTW wafer system should be useful in as many applications as possible, changes were made in the design to increase its run-time programmability. The added features include choice of two distance metrics: the Chebyshev norm as well as the Itakura-Saito log-likelihood metric. The log-likelihood metric is preferred by Bell Laboratories for LPC parameters. The Chebyshev norm has been previously used for spectral weights, cepstral weights, as well as LPC ks. The order of the distance metric has been made run-time programmable to 8, 10, 12 or 16 features. The bit-serial arithmetic approach of the wafer is ideally suited for trading real-time processing capability for arithmetic precision. This has been used to advantage to allow variable precision for feature representation, variable precision for accumulation of the distance metric and programmable truncation of the resulting metric before addition to the accumulated path distance in the path computer.

Two examples of 'operating points' which the DTW wafer system will be able to accommodate are delineated below:

- (1) Eighth-order LPC-based parameters using the Itakura-Saito log-likelihood distance metric:

$$\begin{aligned} D &= 0; \\ \text{for } (0 \leq i \leq P) \\ \{ D &= D + (I[i] * R[i]) \} \\ D &= \log D \end{aligned}$$

where D is the distance for a given input/reference frame comparison, P (the model order) is 8, and $I[i]$ and $R[i]$ (the input and reference frame parameters) are represented to 16 and 12 bits precision, respectively. The summation accuracy of D is 32 bits and the resulting $\log D$ requires only 8 bits of precision.

- (2) 16th order filter-bank-based parameters using the Chebyshev norm:

$$\begin{aligned} D &= 0; \\ \text{for } (0 \leq i \leq P) \\ \{ D &= D + |I[i] - R[i]| \} \end{aligned}$$

where P is 16, $I[i]$ and $R[i]$ are represented to 6 bits precision, the summation precision of D is 16 bits and D is not truncated before delivery to the path computer.

As described in the previous semiannual report, the connected word-recognition system consists of the DTW wafer, the wafer controller and template memory (Figure 2). The wafer functions as a hardware 'subroutine' executing an input-string/reference-word time warp and distance calculation while the external controller handles the control-oriented and less

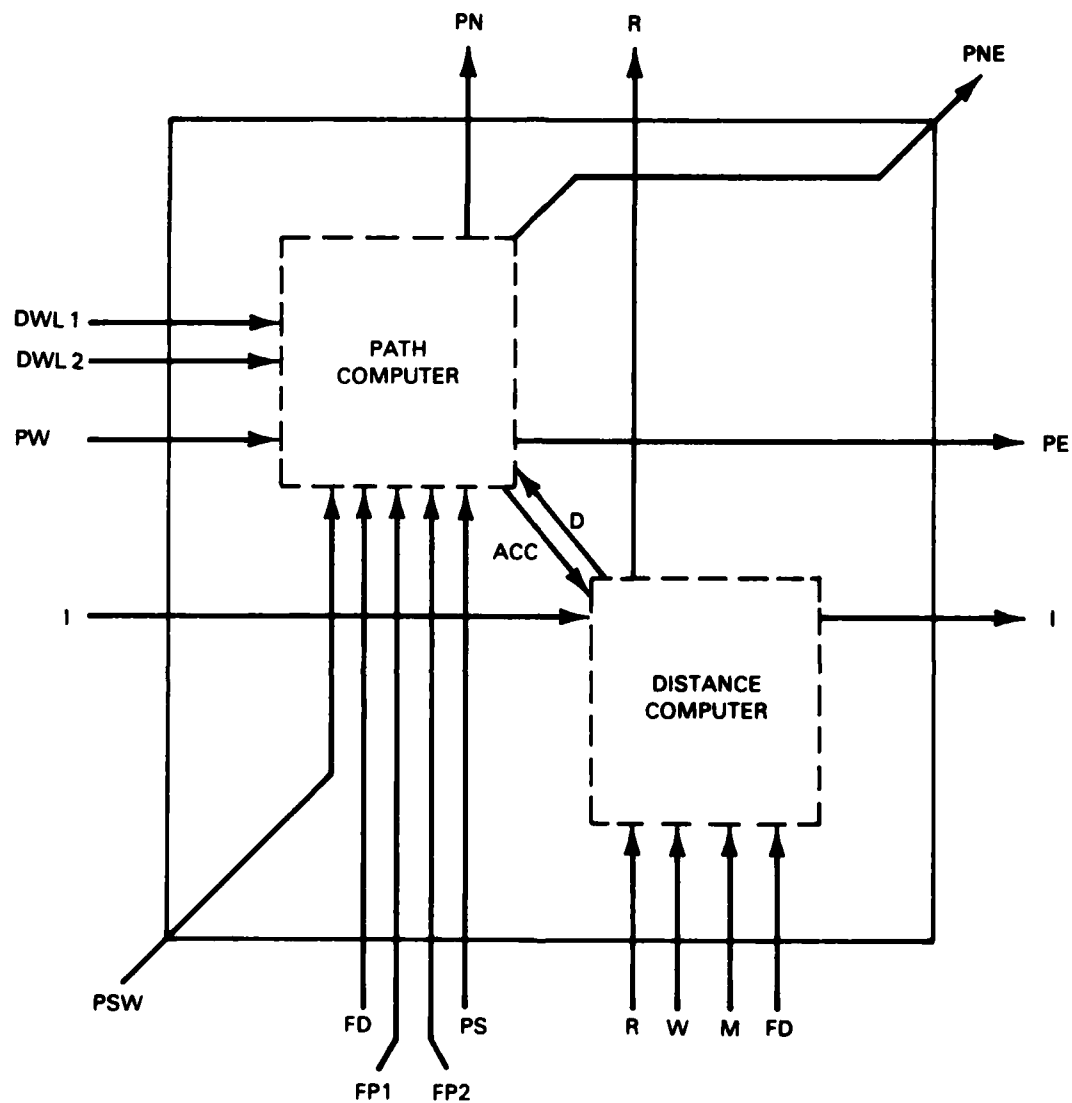


Figure 3. DTW processing element (PE).

135011-N

computationally intensive 'inter-level' tasks requisite to the level-building algorithm. A typical wafer system consists of 63 processing elements (PEs) organized as two diagonals (32 and 31 PEs wide) and 32 delay elements.

The processing element is constructed of a path computer and a distance computer (Figure 3). The simple inter-processor communication resulting from bit-serial processing is dramatically illustrated here: all interconnection is shown except for power and system clock lines. Note that all lines are single bit width.

The distance computer's function is to calculate a dissimilarity or distance measure between an input speech frame and a reference speech frame. The inputs to the computer are the reference and input frame bit-serial streams (R,I), and the word and frame timing signals (W,FD). The distance measure is computed as a sequential summation which is stored in and recalled from the path computer through the ACC and D lines. The static input M is used for run-time choice of either of the two distance metrics described above.

The distance computer (Figure 4) is implemented as a 12-stage pipelined parallel-serial type multiply-accumulate unit preceded by a subtract stage. Also included are input and output buffer stages (DIN and DOUT), metric choice logic (MET) and shift registers for delaying the I and R data before input to the neighboring PEs. One stage of the distance computer is detailed in Figure 5 showing five bits of storage and carry-save adder logic.

The path computer's task is to implement the dynamic time warping (DTW) algorithm by choosing the minimum accumulated path distance from those arriving from its south, southwest, and west, adding the local distance derived from the distance computer and outputting appropriate accumulated path distances to the north, northeast and east. The level-building algorithm 'traceback' pointer must also be routed along with the chosen path distance. The path computer's inputs are the south, southwest and west accumulated distances (PS, PSW, PW), the distance from the distance computer (D) and timing control signals (FP1, FP2, FPD). Its outputs are the north, northeast, and east accumulated distances (PN, PNE, PE) and the partial accumulated distance (ACC) from the distance computer. The accumulated path distance data streams include not only the path distances to that point, but also the 'trackback pointer' information required to implement the level-building aspects of the connected word-recognition algorithm and 'activity bits' used to switch the path computer between an active and initialization mode.

The major circuits in the path computer (Figure 6) are three 32-bit registers to buffer the incoming accumulated path distance (PS, PSW, PW), the bit-serial 3-way minimizer circuit (MIN), bit-serial carry-save adders to add in the local distance D, and 2-way output multiplexers for routing the accumulated paths depending on the mode of the path computer (initialization or active). In addition, there is a run-time programmable delay of D to implement variable accumulation precision and logic controlling the truncation of D before addition to the path distance.

Two types of delay cells are used to compose the delay element of Figure 2: one for the input and reference frame data and one for the accumulated path distances. The input

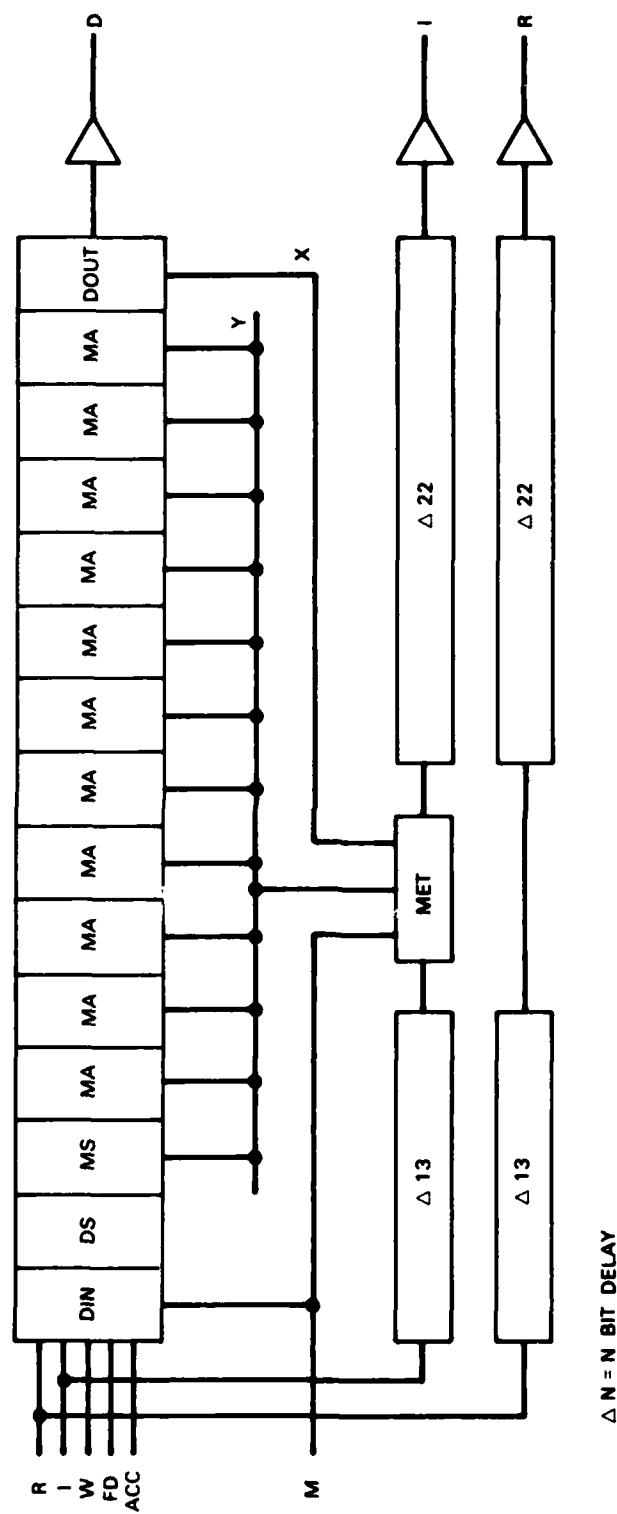


Figure 4. Distance computer.

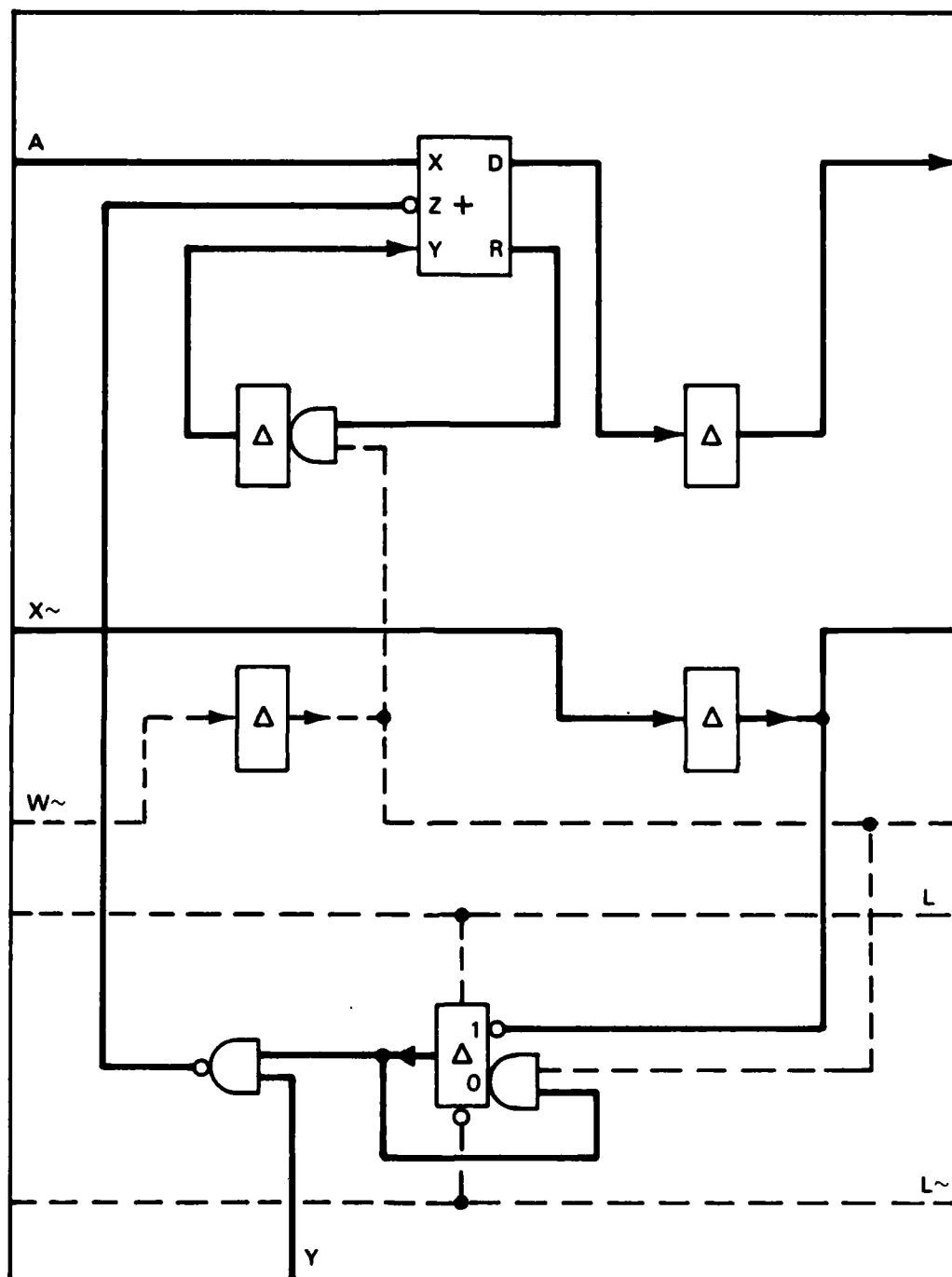


Figure 5. One stage (MA) of distance computer.

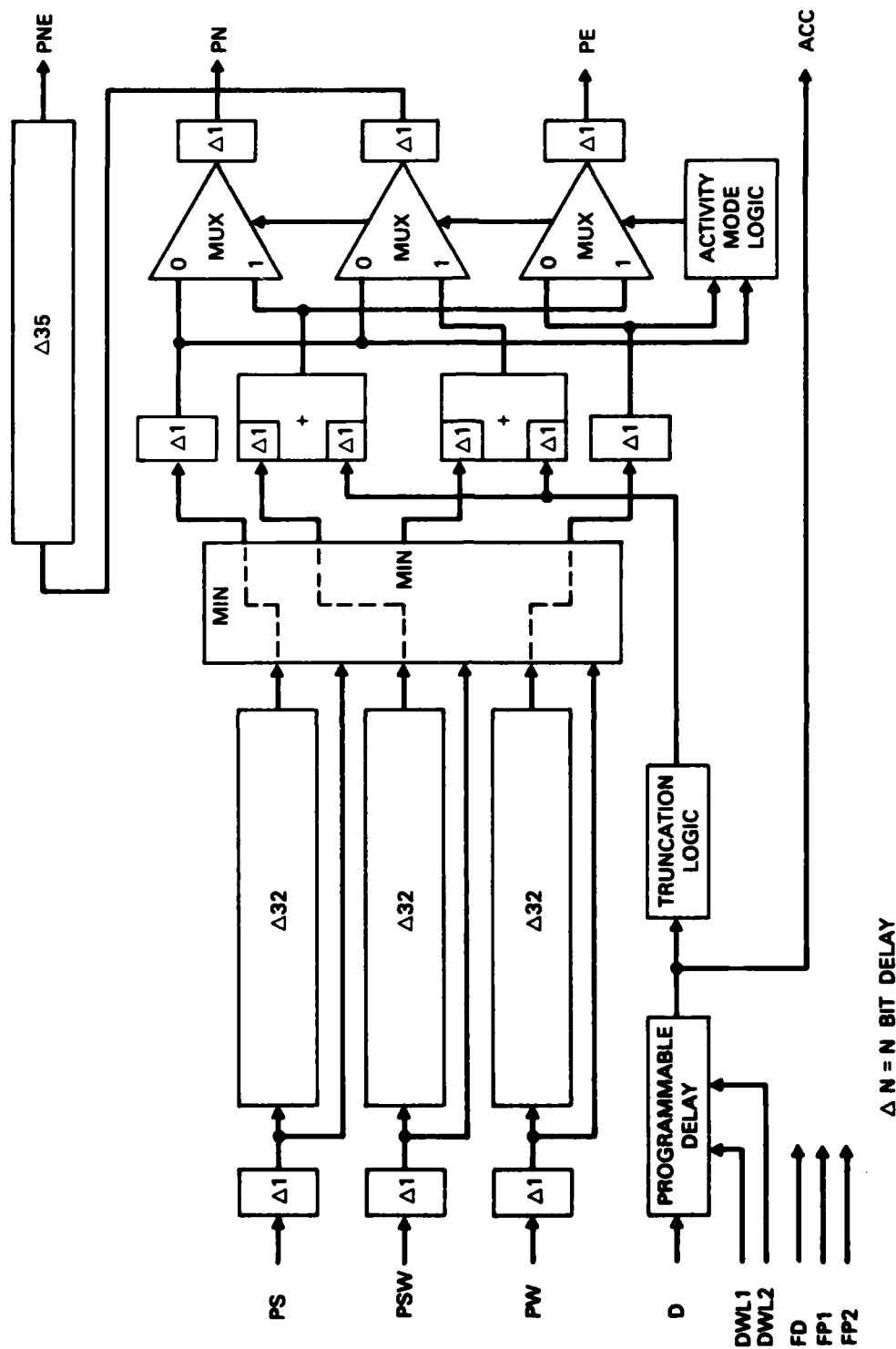


Figure 6. Path computer.

and reference frame data delay length is varied with static control signals DWL 1 and DWL 2 to vary the order of the distance metric from eight to sixteen words.

Circuit layout of the four restructurable elements in the DTW wafer is being done using no butting contacts and only 1-level metal with the Lincoln Laboratory 5- μ M-bulk CMOS design rules. This approach facilitates the option of having wafers fabricated outside Lincoln up through 1st-level metal. A first pass design of the distance computer has been completed at this time. Portions of the remaining three cells have been laid out including the path computer bit-serial three way minimizer. Size estimates have been made on all four cell types indicating that the 63 processor wafers could support a 2:1 circuit redundancy. This is believed to be adequate given the size of the cells and the increased circuit yield expected using 1-level metal processing.

As described above, a C-language program has been written for verification of the detailed digital design of the individual cells as well as the system as a whole. The individual cells have been simulated and wafer system simulations are in progress.

A higher-level program which simulates the entire level-building algorithm has also been written and debugged. An equivalent version of the program has also been demonstrated which uses the Lincoln LDSP (a Laboratory-built high-speed signal processing minicomputer). In this configuration, the host-machine (PDP-11), in conjunction with the appropriate C program, uses the LDSP as a hardware subroutine to implement the inner (and computationally intensive) portion of the level-building algorithm. This approach results in a tenfold decrease in processing time over the totally C-language implementation and is therefore more practical for experimentation where a large amount of data is to be processed. Furthermore, since in this configuration the LDSP simulates the function of the water-scale circuit, it is expected that this will serve as a test and demonstration platform for the final wafer system. The output of this simulation is already being used as a double check against the low-level simulation described above.

B. DIGITAL INTEGRATOR

A major milestone in the fabrication of full-wafer digital integrator systems was the demonstration of the steadily increasing cell yield at first-level metal test to greater than 50%, as shown in Figure 7, comfortably above 33% required. Further attrition between first- and second-level metal was traced to inadequate coverage of polysilicon contacts during amorphous silicon etch. Mask changes have been implemented to eliminate this problem and negligible attrition was experienced after first level metal in subsequent wafers. Partially-populated test wafers and the first full wafer are in programming now. Four more full wafers will be ready for programming in October.

The 68000 restructurability tester is now completed and interfaced to the VAX. This allows down-loading of test vector files from the VAX and testing of individual cells during the restructuring process as they are linked into the system.

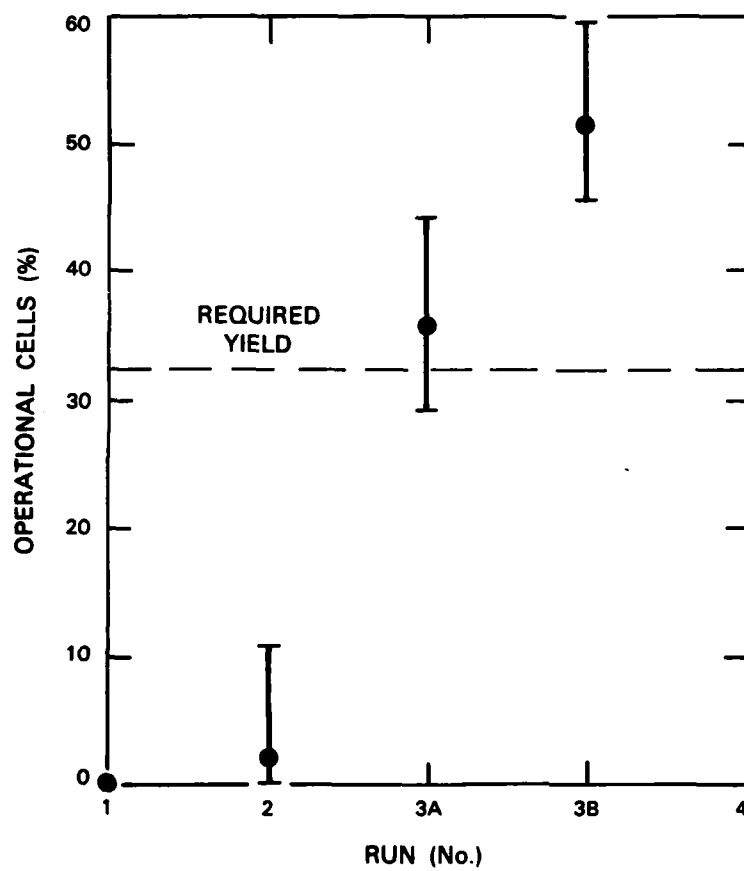


Figure 7. Digital integrator yields.

131038-N-01

IV. TESTING

A. TESTER-ON-CHIP (TOC) DEVELOPMENT

The TOC system is a low-cost functional IC tester, consisting of an array of four-bit slices, together with a small amount of common interface and control circuitry. For testing dynamic circuitry, there is a provision for looping through a hold sequence, using one memory bank while the other is being reloaded. Comprising about 4000 transistors, the TOC element represents the most ambitious design yet undertaken with the MACPITTS silicon compiler.

Thirteen TOC chips were received from the M33M MOSIS 3-micron NMOS test run at the end of June. They were promptly tested on the Tektronix 3260, using a pattern file generated by the nl switch level simulation used to verify the design prior to fabrication. Unfortunately, none of them worked, and only three showed any signs of life whatsoever. This prompted an optical inspection, which revealed incomplete etching of the metalization layer. On all but the three chips that showed signs of life, the clock distribution lines were observed to be shorted together. On all chips, shorts were noted in the Weinberger array control section, and the pad wiring channels. Figure 8 contains photomicrographs of these areas.

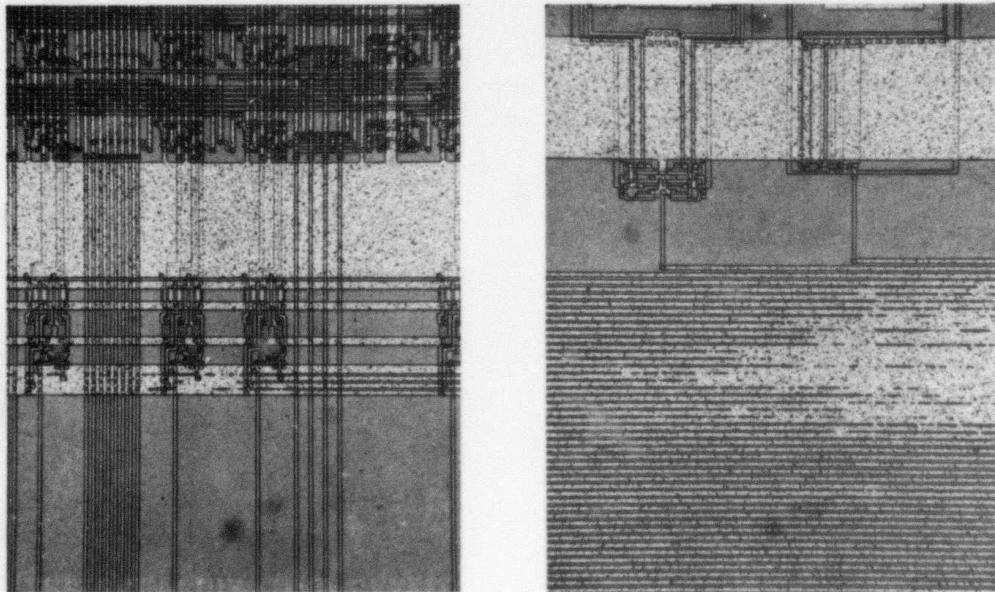
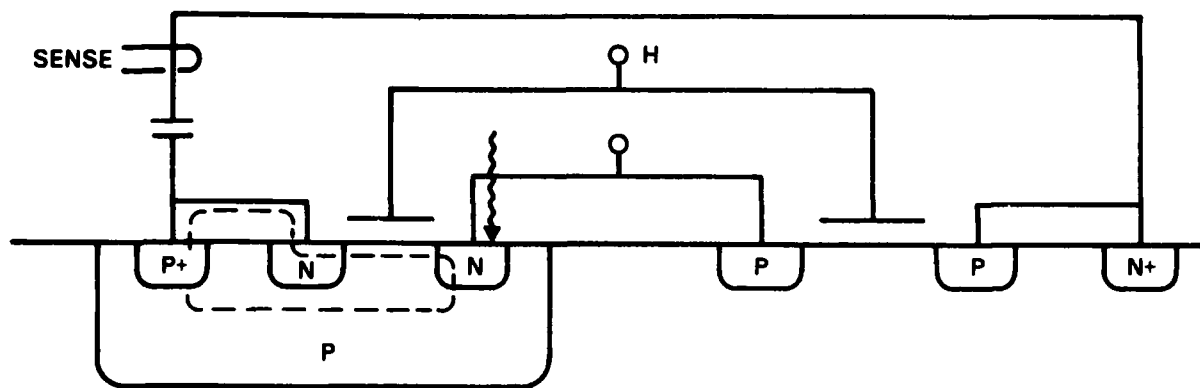


Figure 8. Photomicrographs of clock distribution wires and pad wiring channels in defective TOC circuits.

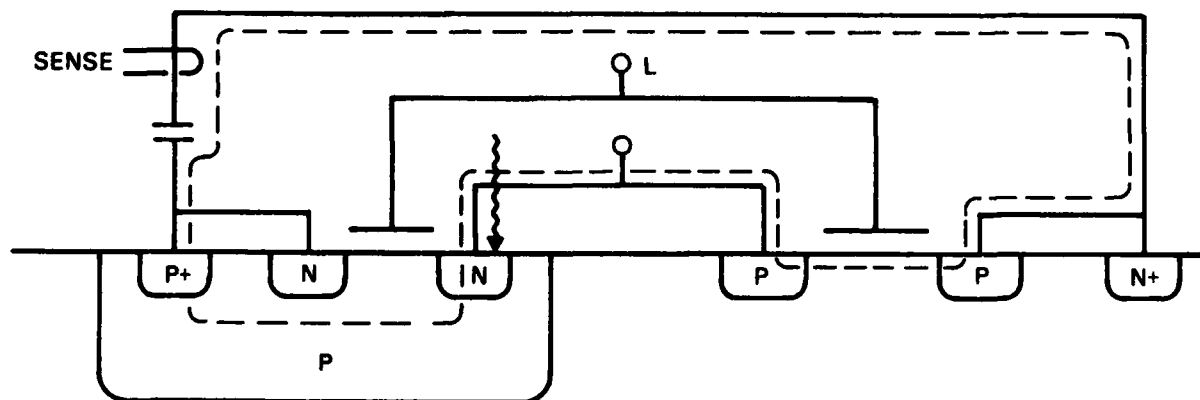
Because of the changes to MACPITTS during the development of TOC, and especially on the organelle library, a second, simple test design was submitted on M33M. This chip consisted of a 4-bit register, an incrementer, and an xor unit, along with some control lines, and an I/O port. Four were received along with the TOC chips and, after the problems were discovered, were tested in the same way, with the Tektronix 3260 and a pattern file generated by an nl simulation. All worked, and half were seen to operate at the 5-MHz maximum tester speed for this mode. Optical inspection confirmed the lack of metalization defect. The only explanation is that this design is very small compared to TOC.

B. EXPERIENCE WITH THE OPTICAL PROBE

When the drain junction of an n-channel transistor in a CMOS inverter circuit is illuminated, the photo electrons will flow through the n-channel transistor if it is conducting and return to the junction as shown in Figure 9a. If the n-channel transistor is turned off and the circuit is working properly, then the current will flow through the conducting p-channel transistor and return to the illuminated junction through the power supply as shown in Figure 9b. Therefore, by monitoring the current in the power supply connection, the state of the inverter, and, similarly, any level restoring logic gate, can be determined. The same type of measurement can be made with NMOS circuits but back-gate bias effects make it more complicated than with CMOS [8]. On a MOSIS-fabricated bulk CMOS 74181-equivalent circuit, a 30:1 ratio of photo current in the power line was measured between the ZERO and ONE states of correctly operating logic gates. On defective gates, intermediate signal levels were observed. Further work is required to determine the behavior of faulty circuits. For this experiment, a laser restructuring setup was used but the required equipment is very simple: a mechanically modulated low-power laser, a microscope with stage, and an oscilloscope and simple current-monitoring circuit. For larger circuits, it is very convenient to have a computer-controlled stage and TV monitor.



(a)



(b)

Figure 9. Current paths for laser probing of a CMOS inverter: (a) N-channel conducting, (b) P-channel conducting.

V. RVLSI TECHNOLOGY

The Restructurable Weinberger Array (RWEIN) is a demonstration of intracell restructurability, using a simple addition to the standard MOSIS NMOS process: a layer of polyimide. This material, when pyrolysed by a laser pulse, forms a conducting carbon deposit with a typical resistance of less than 1.5 k Ω . We use this mechanism (polyimide link) to connect adjacent metal lines [9].

The demonstration vehicle chosen is a 64-by-4-gate Weinberger-style array. It is customized to a particular graph of nor gates by making polyimide links, and cutting metal lines.

The processing of the first pieces of wafers has been finished. Initial experiments on test structures have identified the important parameters for making links and cutting lines. These include power levels, spot sizes and positions. These experiments verified our ability to make connections with 1- to 2-k Ω resistances. The line-cutting experiments were also successful.

Figure 10 is a symbolic representation of the layout of a 4-bit ALU slice for the restructurable Weinberger array. The symbols are decoded as follows:

- Represents a horizontal through way.
- * The output from this column is tapped onto the track in both directions.
- v This column will be pulled down by this track from either direction.
- blank There is no connection of the track across this column; however, vertical conductivity is maintained.
- (The output from this column is tapped onto the track to the left.
-) The output from this column is tapped onto the track to the right.
- / This column will be pulled down by this track from the right.
- < This column will be pulled down from the right, and the output will be tapped off to the left.
- > This column will be pulled down from the left, and the output will be tapped off to the right.

The rows between active element rows are special, and they are decoded as follows:

- + This represents a pull-up, identifying the top of a nor gate.
- | This represents a vertical through way.
- blank A blank in this context implies a break in the vertical connectivity.

This layout has been fully simulated, using a program to duplicate the action of making and breaking connections in the cif and nl data bases. Programming of the array is done in two phases. First, all the gate and drain links of the pull-down sites that are to be used are made. Then the array is probed, and each track is turned on in sequence. By observing

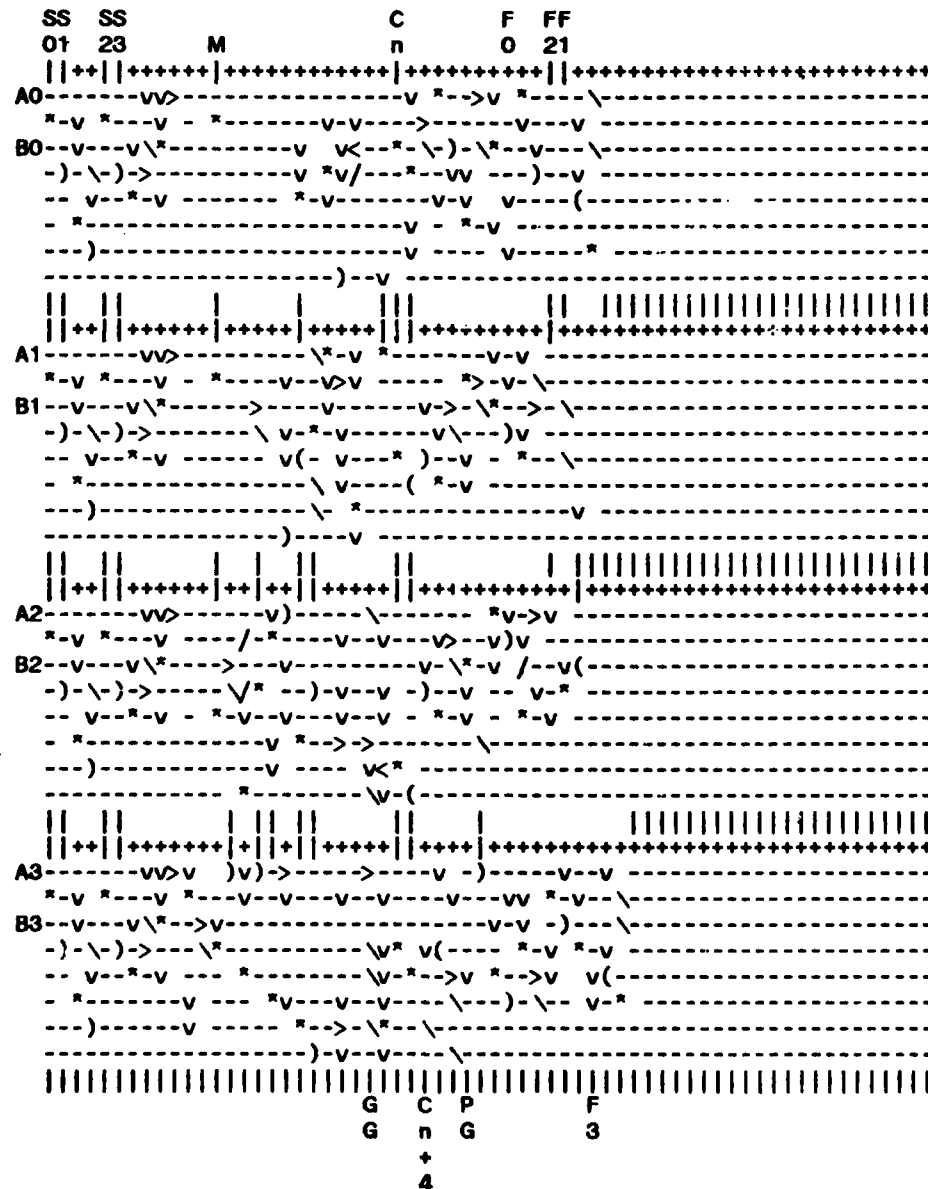


Figure 10. Symbolic representation of 4-bit ALU slice layout.

the voltage drop on each column, all the transistors used in the circuit are tested. Then, during phase 2, the output taps and segmentation cuts are made, producing the finished product.

The first cells of the first wafer have been programmed with phase 1 of this circuit. These have been probed and a systematic problem has surfaced. The odd-numbered columns exhibit up to 30% of the links with abnormally high resistance whereas the even-numbered columns have been 100% normal. So far, there has been no success at eliminating the unreliability in links made in the odd-numbered columns of the restructurable Weinberger array. Several attempts have been made to eliminate all the dissimilarities between even and odd columns, resulting from asymmetry in the layout and alignment of the polyimide mask.

A new optical head for the laser table is being installed which has movable aperture blades. This will permit the use of a spot with a variable aspect ratio. Our experience is that the most reliable links are made with a single laser zap large enough to cover the metal on both sides of the link. This has not been possible because a sufficiently large spot is so big that it extends into adjacent areas. The variable aperture will allow us to use an ellipsoidal spot that reaches both metal pieces, but does not extend into unwanted areas.

REFERENCES

1. J.M. Siskind, J.R. Southard, and K.W. Crouch, "Generating Custom High Performance VLSI Designs From Succinct Algorithmic Descriptions," *Proc. Conf. on Advanced Research in VLSI*, Cambridge, Massachusetts (25-27 January 1982), pp. 28-40.
2. C.M. Fiduccia and R.M. Matheyses, "A Linear-Time Heuristic for Improving Network Partitions," 19th Design Automation Conf. (IEEE Computer Soc. Press, 13 June 1982), pp. 175-181.
3. Restructurable VLSI Program Semiannual Technical Summary, Lincoln Laboratory, M.I.T. (31 March 1983), DTIC AD-A131246.
4. J. Soukup, "Circuit layout," *IEEE Proc.* **69**, 1281-1304 (October 1981).
5. J.K. Ousterhout, "CAESAR: An Interactive Editor for VLSI Layout," *VLSI Design* (Fourth Quarter, 1981).
6. W.S. Chan, "A New Channel Router Algorithm," *Proc. 3rd Caltech Conf. on VLSI*, edited by Randal Bryant (Computer Science Press, 1983), pp. 117-139.
7. C.S. Myers and L.R. Rabiner, "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition," *IEEE Trans. Acoust., Speech, and Signal Processing* **ASSP-29**, 284-297 (April 1983).
8. Quarterly Technical Summary, Advanced Electronic Technology, Lincoln Laboratory, M.I.T. (15 February 1981), DTIC AD-B074592-L.
9. J.I. Raffel, J.F. Freidin, and G.H. Chapman, "Laser-formed Connections Using Polyimide," *Appl. Phys. Lett.* **42**, 705-706 (15 April 1983).

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-83-104	2. GOVT ACCESSION NO. AD-A139 959	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Restructurable VLSI Program		5. TYPE OF REPORT & PERIOD COVERED Semiannual Technical Summary 1 April — 30 September 1983
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Peter E. Blankenship		8. CONTRACT OR GRANT NUMBER(s) F19628-80-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M.I.T. P.O. Box 73 Lexington, MA 02173-0073		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Order 3797 Program Element No. 61101E Project No. 3D30
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, VA 22209		12. REPORT DATE 30 September 1983
		13. NUMBER OF PAGES 38
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Electronic Systems Division Hanscom AFB, MA 01731		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES None		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
VLSI Restructurable VLSI (RVLSI) programmable interconnect defect avoidance	customization hardware description language placement routing	systolic array integrator wafer-scale systems speech recognition
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
<p>This report describes work performed on the Restructurable VLSI Program sponsored by the Information Processing Techniques Office of the Defense Advanced Research Projects Agency during the semiannual period 1 April — 30 September 1983.</p>		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)